OT-Flow: Optimal Transport for Continuous Normalizing Flows

Derek Onken§, Samy Wu Fung†, Xingjian Li§, Lars Ruthotto§

§Departments of Mathematics and Computer Science at Emory University, †Department of Mathematics at UCLA

Motivation

Continuous Normalizing Flows

A normalizing flow [1] is an invertible mapping $f\colon \mathbb{R}^d \to \mathbb{R}^d$ between an arbitrary probability distribution and a standard normal distribution whose densities we denote by ρ_0 and ρ_1, respectively.

By change of variables, the flow must approximately satisfy

$$\log \rho_0(\boldsymbol{x}) = \log \rho_1(f(\boldsymbol{x})) + \log|\det \nabla f(\boldsymbol{x})| \quad \text{for all} \quad \boldsymbol{x} \in \mathbb{R}^d. \qquad (1)$$

In continuous normalizing flows (CNFs), f is obtained by solving the neural ordinary differential equation (ODE) [2, 3]

$$\partial_t \begin{bmatrix} \boldsymbol{z}(\boldsymbol{x},t) \\ \ell(\boldsymbol{x},t) \end{bmatrix} = \begin{bmatrix} \mathbf{v}(\boldsymbol{z}(\boldsymbol{x},t),t;\boldsymbol{\theta}) \\ \mathrm{tr}\left(\nabla \mathbf{v}(\boldsymbol{z}(\boldsymbol{x},t),t;\boldsymbol{\theta})\right) \end{bmatrix}, \quad \begin{bmatrix} \boldsymbol{z}(\boldsymbol{x},0) \\ \ell(\boldsymbol{x},0) \end{bmatrix} = \begin{bmatrix} \boldsymbol{x} \\ 0 \end{bmatrix}, \qquad (2)$$

where, for artificial time $t \in [0,T]$,

- $\boldsymbol{x}$ maps to $f(\boldsymbol{x}) = \boldsymbol{z}(\boldsymbol{x},T)$ following trajectory $\boldsymbol{z}\colon \mathbb{R}^d \times [0,T] \to \mathbb{R}^d$
- dynamics are modeled by neural network layer $\mathbf{v}\colon \mathbb{R}^d \times [0,T] \to \mathbb{R}^d$ parameterized by weights $\boldsymbol{\theta}$
- $\ell(\boldsymbol{x},T) = \log\det \nabla f(\boldsymbol{x})$, derived from Jacobi's Formula [2]

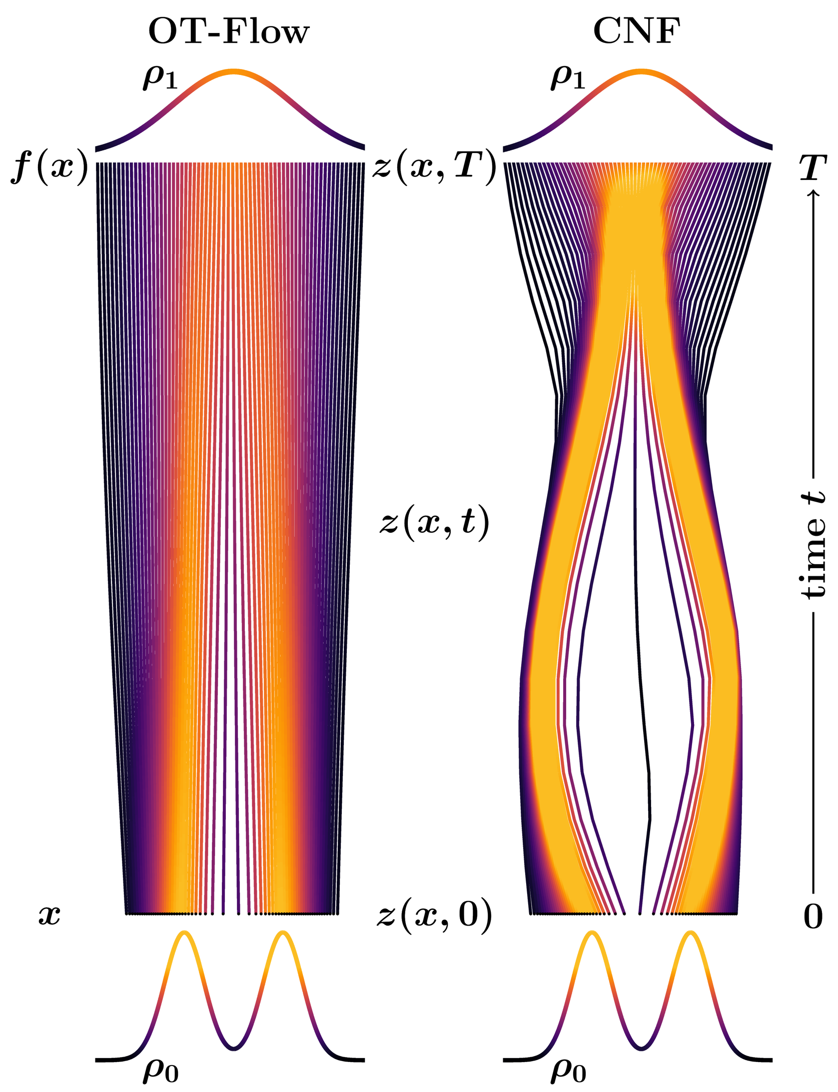


Figure 1: While a CNF can have curved trajectories, OT-Flow's are straight (modification of Fig. 1 in [3, 4]).

Microscale: an arbitrary sample $\boldsymbol{x}$ maps to a normally distributed $f(\boldsymbol{x})$
Macroscale: ρ_0 maps to ρ_1

CNFs are trained by solving the optimization problem [1, 3]

$$\min_{\boldsymbol{\theta}} \ \mathbb{E}_{\rho_0(\boldsymbol{x})} \left\{ C(\boldsymbol{x},T) := \frac{1}{2}\|\boldsymbol{z}(\boldsymbol{x},T)\|^2 - \ell(\boldsymbol{x},T) + \frac{d}{2}\log(2\pi) \right\} \quad \text{s.t.} \quad (2). \qquad (3)$$

High Training Costs

- Many functions evaluations are needed to solve (2)
- Using automatic differentiation (AD) to compute the trace requires separate matrix-vector products with the Jacobian and all d standard basis vectors, costing $\mathcal{O}(d^2)$ FLOPs in total

Our Contributions

Optimal Transport (OT) Incorporating OT, we regularize the CNF so it has a unique solution (Figure 1).

Analytic Exact Trace We derive formulae for an exact trace computation with complexity $\mathcal{O}(d)$ FLOPs.

Optimal Transport

L_2 Transport Costs Add transport costs

$$L(\boldsymbol{x},T) = \int_0^T \frac{1}{2}\|\mathbf{v}(\boldsymbol{z}(\boldsymbol{x},t),t)\|^2 \, \mathrm{d}t. \qquad (4)$$

to (3) to penalize the arc-length of the trajectories.

Potential Function By the Pontryagin maximum principle [5], there exists a scalar potential function $\Phi\colon \mathbb{R}^d \times [0,T] \to \mathbb{R}$ such that

$$\mathbf{v}(\boldsymbol{x},t;\boldsymbol{\theta}) = -\nabla\Phi(\boldsymbol{x},t;\boldsymbol{\theta}).$$

Idea: Analogous to classical physics, samples move to minimize their potential
⇒ We parameterize potential Φ instead of $\mathbf{v}$.

HJB Regularizer At optimality, Φ satisfies the Hamilton-Jacobi-Bellman (HJB) equation [6]

$$-\partial_t \Phi(\boldsymbol{x},t) + \frac{1}{2}\|\nabla\Phi(\boldsymbol{z}(\boldsymbol{x},t),t)\|^2 = 0,$$
$$\Phi(\boldsymbol{x},T) = G(\boldsymbol{x})$$

where

$$G(\boldsymbol{z}(\boldsymbol{x},T)) = 1 + \log\left(\rho_0(\boldsymbol{x})\right) - \log\left(\rho_1(\boldsymbol{z}(\boldsymbol{x},T))\right) - \ell(\boldsymbol{x},T)$$

To penalize sub-optimality, use HJB regularizer

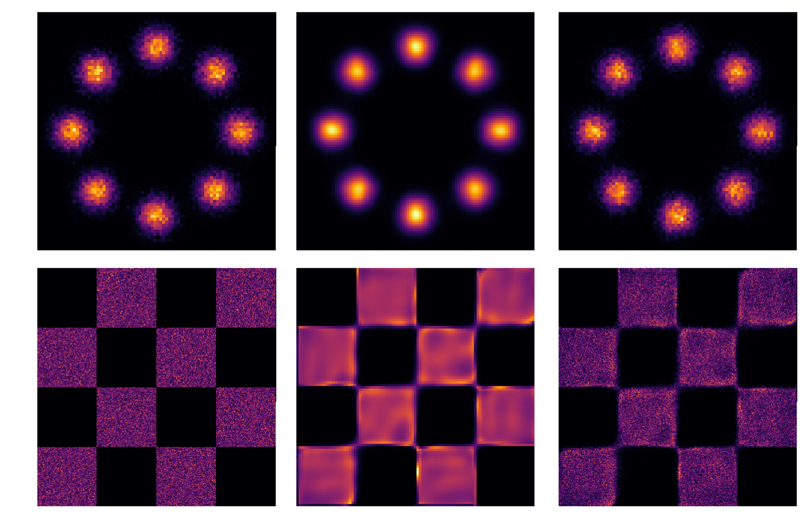
$$R(\boldsymbol{x},T) = \int_0^T \left| \partial_t\Phi(\boldsymbol{z}(\boldsymbol{x},t),t) - \frac{1}{2}\|\nabla\Phi(\boldsymbol{z}(\boldsymbol{x},t),t)\|^2 \right| \mathrm{d}t \qquad (5)$$

OT-Flow Optimization Problem

$$\min_{\boldsymbol{\theta}} \ \mathbb{E}_{\rho_0(\boldsymbol{x})} \left\{ C(\boldsymbol{x},T) + L(\boldsymbol{x},T) + R(\boldsymbol{x},T) \right\} \ \text{s.t. (2), (4), (5)}$$

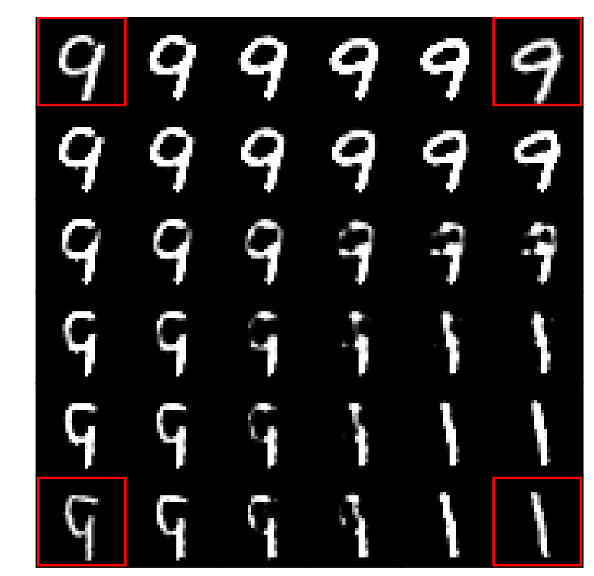
Results

Two-dimensional Toy Density Estimation



19x speedup in training and 28x speedup in inference relative to the state-of-the-art FFJORD [3] on five real datasets of dimensionality $d = 6, 8, 21, 43, 63$

MNIST Synthetic Generation



Encoder $B\colon \mathbb{R}^{784} \to \mathbb{R}^{128}$,
Decoder $D\colon \mathbb{R}^{128} \to \mathbb{R}^{784}$
such that $D(B(\boldsymbol{x})) \approx \boldsymbol{x}$

Train OT-Flow mapping samples $B(\boldsymbol{x})$ to $\mathcal{N}(0, \boldsymbol{I}_{128})$

For two MNIST images $\boldsymbol{x}_1, \boldsymbol{x}_2 \in \mathbb{R}^{784}$, interpolate in the latent space to create synthetic image

$$\boldsymbol{y}(\lambda) = D(f^{-1}(\lambda f(B(\boldsymbol{x}_1)) + (1-\lambda) f(B(\boldsymbol{x}_2)))).$$

Original images are boxed in red.

Links

Corresponding Paper Preprint:

arxiv.org/abs/2006.00104

PyTorch implementation is available on Emory's Machine Learning and Inverse Problems repository:

github.com/EmoryMLIP/OT-Flow

Acknowledgments

Emory Funding: National Science Foundation awards DMS 1522599 and CAREER DMS 1751636, Binational Science Foundation Grant 2018209, and NVIDIA Corporation

UCLA Funding: AFOSR MURI FA9550-18-1-0502, AFOSR Grant No. FA9550-18-1-0167, and ONR Grant No. N00014-18-1-2527.

Special Thanks: IPAM Long Program MLP 2019 organizers & staff

References

[1] D Rezende and S Mohamed *Variational Inference with Normalizing Flows.* ICML, 2015.

[2] Chen et al. *Neural Ordinary Differential Equations.* NeurIPS, 2018.

[3] W Grathwohl et al. *FFJORD: Free-form continuous dynamics for scalable reversible generative models.* ICLR, 2019.

[4] C Finlay et al. *How to train your neural ODE.* arXiv:2002.02798, 2020.

[5] L Evans *An Introduction to Mathematical Optimal Control Theory Version 0.2.* 2013.

[6] L Evans *Partial differential equations and Monge-Kantorovich mass transfer.* Current developments in mathematics, 1997.

Exact Trace

Exploit $\mathrm{tr}\left(\nabla^2\Phi(\boldsymbol{s};\boldsymbol{\theta})\right) = \mathrm{tr}\left(\boldsymbol{E}^\top \nabla^2_{\boldsymbol{s}}\Phi(\boldsymbol{s};\boldsymbol{\theta})\,\boldsymbol{E}\right)$ for $\boldsymbol{E} =$ `eye(d+1,d)`. Details in Links.

In runtime, exact trace is competitive with estimators used in other CNFs.